

## Re: why microsoft choose mfc rather than wtl?

---

*Source:* <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.mfc/2005-04/msg00748.html>

---

- *From:* Joseph M. Newcomer <[newcomer@xxxxxxxxxxxxx](mailto:newcomer@xxxxxxxxxxxxx)>
  - *Date:* Sat, 09 Apr 2005 12:38:26 -0400
- 

"Neat" is not a good reason to adopt a technology for mainstream programming. I've abandoned a lot of "neat" technologies over the years because that was their only virtue.

Small size? You had better have a really compelling excuse about why this makes sense. I find that it rarely matters; this is a holdover from the days of 64K PDP-11s, and the attitude about the "huge preinstalled DLL" isn't really viable today; I find the arguments very non-compelling in an era of 2GHz machines, 10000RPM disk drives, DSL, and 1GB memories on consumer machines. I remember worrying about those things, thirty years ago. But this is 2005, and the metrics of quality and utility are quite different today.

I've had enough trouble convincing clients that MFC and C++ are state-of-the art (to the last one who was reluctant to accept this new, daring technology, in 1998, I basically said, "It will cost you k dollars and n months to product this in MFC. It will cost you 4\*k dollars and 2\*n months to produce it in C. Part of the reason for the cost change is that my hourly rate doubles to compensate me for pain and suffering". We did the project in n/2 months at 0.8k dollars, and they now believe). But I would not dare propose something that was apparently undocumented, difficult to learn, and difficult to extend; they have to live with this stuff for years.

If I can't create a control subclass in under a minute, I don't want to hear about it. I'm always extending controls. I did three new controls this morning (one each derived from CEdit, CListCtrl and CComboBox). Total time, including fully functioning code, was less than 30 minutes each.

The reason for ATL is twofold: first, to make controls small enough to put on a Web page. This is a bogus reason, because putting native executable code on a Web page to implement a control is deeply unethical. This is a technology beloved by crackers and cyberterrorists, and has no place in modern computing. I have three levels of firewalls, each of which is programmed to remove all ActiveVirus controls (funny, most of the IE security problems reported are nonexistent if you simply turn client-side scripting off! Hasn't anyone figured out yet that we have to Just Say No?). The other reason is so that controls could be built that could be used by C, C++ (no MFC), MFC, VB, Delphi, and as far as I know, COBOL and FORTRAN, without requiring a complete runtime that only supported one of those environments. But for an application, this excuse doesn't exist).

joe

On Sat, 09 Apr 2005 15:37:39 +0100, Daniel James <[wastebasket@xxxxxxxxxxxxxxxxx](mailto:wastebasket@xxxxxxxxxxxxxxxxx)> wrote:

Re: why microsoft choose mfc rather than wtl?

>In article news:<uhad5117gqaohjo544ig47m9l6m7mh69bg@xxxxxxx>, Joseph M. Newcomer  
>wrote:  
>> What are the advantages of WTL (which I keep hearing about but can't  
>> find any documentation about) over MFC (which has massive amounts of  
>> documentation)?  
>  
>WTL is kinda neat ... it's very lightweight and can produce very small executables  
>(without relying on a huge preinstalled DLL to achieve this, the way that MFC apps do  
>to).  
>  
>It doesn't have the broad range of features that MFC does, but it does provide  
>wrappings for wrap most of the GUI parts of Win32 that MFC wraps (and does so in  
>similar ways, so knowledge of MFC will help). The project is now run on SourceForge at  
><http://sourceforge.net/projects/wtl/> ... but there's no documentation that I can see  
>.. Google will find some WTL documentation projects on the web, though.  
>  
>One problem with WTL is that it uses C++ templates very heavily (or, as heavily as one  
>can with an elderly compiler like VC6 ... the current version may require VC7, I'm not  
>sure) and so is not for the beginner or the faint-hearted. It's definitely much harder  
>to extend/modify than MFC. I've written a few short apps in WTL and found it very  
>usable ... but I've never dared use it for a large client project (partly because I  
>have no confidence that my clients (or their customers) would be able to maintain the  
>heavily-templated code after I've moved on). It's no harder, really, than using ATL  
>for controls, though.  
>  
>Cheers,  
> Daniel.  
>  
>  
>

Joseph M. Newcomer [MVP]  
email: [newcomer@xxxxxxxxxxxxx](mailto:newcomer@xxxxxxxxxxxxx)  
Web: <http://www.flounder.com>  
MVP Tips: [http://www.flounder.com/mvp\\_tips.htm](http://www.flounder.com/mvp_tips.htm)  
.

---

• *Follow-Ups:*

- ◆ ***Re: why microsoft choose mfc rather than wtl?***  
    ◇ From: Mark Randall
- ◆ ***Re: why microsoft choose mfc rather than wtl?***  
    ◇ From: Daniel James

• *References:*

- ◆ ***why microsoft choose mfc rather than wtl?***  
    ◇ From: Huang Shu Huai
- ◆ ***Re: why microsoft choose mfc rather than wtl?***  
    ◇ From: Joseph M . Newcomer
- ◆ ***Re: why microsoft choose mfc rather than wtl?***

Re: why microsoft choose mfc rather than wtl?

◇ *From:* Daniel James

- Prev by Date: [\*\*Re: visual studio c++ .net 2005 and mfc?\*\*](#)
- Next by Date: [\*\*Re: CSocket, best way to wait for a reply?\*\*](#)
- Previous by thread: [\*\*Re: why microsoft choose mfc rather than wtl?\*\*](#)
- Next by thread: [\*\*Re: why microsoft choose mfc rather than wtl?\*\*](#)
- Index(es):
  - ◆ [\*\*Date\*\*](#)
  - ◆ [\*\*Thread\*\*](#)